UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/593,375	02/26/2007	Yousuke Sakao	20266	1278

23389 7590 06/21/2010
SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
COLUCCI, MICHAEL C	
ART UNIT	PAPER NUMBER
2626	
MAIL DATE	DELIVERY MODE
06/21/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/593,375	**Applicant(s)** SAKAO ET AL.	
	Examiner MICHAEL C. COLUCCI	**Art Unit** 2626	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 September 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 12-22 are rejected under 35 U.S.C. 101 because:

Claims 12-22 do not fall within one of the four statutory categories of invention. Supreme Court precedent[1] and recent Federal Circuit decisions[2] indicate that a statutory "process" under 35 U.S.C. 101 must (1) be tied to another statutory category (such as a particular apparatus), or (2) transform underlying subject matter (such as an article or material) to a different state or thing. While the instant claim(s) recite a series of steps or acts to be performed, the claim(s) neither transform underlying subject matter nor positively tie to another statutory category that accomplishes the claimed method steps, and therefore do not qualify as a statutory process.

Claims 12-22 recite purely mental steps and would not qualify as a statutory process. In order to qualify as a statutory process, the method claim should positively recite the other statutory class to which it is tied (i.e. apparatus, device, product, etc.). For example, the method steps of claim 12 appear to recite mental steps such as "generating a sentence structure… determining patterns" and do not identify an

[1] *Diamond v. Diehr*, 450 U.S. 175, 184 (1981); *Parker v. Flook*, 437 U.S. 584, 588 n.9 (1978); *Gottschalk v. Benson*, 409 U.S. 63, 70 (1972); *Cochrane v. Deener*, 94 U.S. 780, 787-88 (1876).

[2] *In re Bilski*, 88 USPQ2d 1385 (Fed. Cir. 2008).

apparatus that performs the recited method steps, such as a computer executing the processing of text data as described in the specification (present invention spec page 10).

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Akers et al. US 6278967 B1 (hereinafter Akers) in view of Kumai US 20040260979 A1 (hereinafter Kumai).

Re claims 1 and 12, Kumai teaches text mining apparatus comprising:

means for generating a sentence structure from an input document ([0062] & Fig. 5);

means for generating a similar structure of patterns having a similar meaning of a partial structure of the sentence structure by performing predetermined conversion operation ([0062] & Fig. 5),

including at least change in connection of branches in a graph structure, of the partial structure; and means for determining the patterns having the similar meaning as the identical pattern and detecting the pattern ([0062] & Fig. 5)

However, Kumai fails to teach including at least change in connection of branches in a graph structure, of the partial structure.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate including at least change in connection of branches in a graph structure, of the partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 2 and 13, Kumai teaches a text mining apparatus according to Claim 1, further comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]); and

an analyzing unit that inputs and analyzes the document of the storage unit and obtains the sentence structure ([0062] & Fig. 5),

wherein the analyzing unit analyzes the document, and generates the sentence structure ([0062] & Fig. 5) containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee.

However, Kumai fails to teach a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 3 and 14, Kumai teaches a text mining apparatus according to Claim 1, wherein the means for generating the similar structure comprises:

means for performing parallel modification of the sentence structure; means for generating a partial structure of the sentence structure ([0062] & Fig. 5);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary ([0062] & Fig. 5); and

the means for generating the similar structure uses the similar structures as an equivalent class of the partial structure of the sentence structure ([0062] & Fig. 5)

However, Kumai fails to teach means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 4, 15, and 23, Kumai teaches a text mining apparatus comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]);

an analyzing unit that reads and analyzes the document from the storage unit and obtains the sentence structure ([0062] & Fig. 5);

a similar-structure generating unit that performs predetermined modification operation ([0062] & Fig. 5),

including at least change in connection of branches in a graph structure,

of the partial structure of the sentence structure obtained by the analysis of the analyzing unit, and generates a similar structure of patterns having a similar meaning ([0062] & Fig. 5); and

a pattern detecting unit that uses the similar structure generated by the similar-structure generating unit as an equivalent class of the partial structure on the generation source, and detects the pattern ([0062] & Fig. 5)

However, Kumai fails to teach including at least change in connection of branches in a graph structure, of the partial structure.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate including at least change in connection of branches in a graph structure, of the partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 5 and 16, Kumai teaches a text mining apparatus according to Claim 4, wherein the pattern detecting unit uses the similar structure as the equivalent class of the partial structure on the generation source, and detects the pattern ([0062] & Fig. 5).

Re claims 6 and 17, Kumai teaches a text mining apparatus according to Claim 4, wherein the similar-structure generating unit comprises:

means for performing parallel modification of the sentence structure ([0062] & Fig. 5);

means for generating a partial structure of the sentence structure ([0062] & Fig. 5);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary ([0062] & Fig. 5); and the similar-structure generating unit generates the similar structure of the sentence structure and sets the similar structure as an equivalent class ([0062] & Fig. 5).

However, Kumai fails to teach means for performing non-ordering of ordering trees in the sentence structure and/or partial structure

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for performing non-ordering of ordering trees in the sentence structure and/or partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 7 and 18, Kumai teaches a text mining apparatus ([0008]) according to Claim 4, further comprising:

However, means for adjusting the operation so that a user determines how similar patterns are identical and detecting the pattern

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate adjusting the operation so that a user determines how similar patterns are identical and detecting the pattern as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 8, 19, 24, and 25, Kumai teaches a text mining apparatus comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]);

an analyzing unit that reads and analyzes the document from the storage unit and obtains a sentence structure ([0062] & Fig. 5);

whether or not the structures are identical one every type of differences between the sentence structures ([0062] & Fig. 5);

whether or not the structures are identical ones every type of differences between attribute values ([0062] & Fig. 5);

a similar-structure generating unit that performs predetermined conversion operation of a partial structure of the sentence structure obtained by the analyzing unit in accordance with the first determination item generated by the similar-structure generation adjustment unit and generates similar structures having a similar meaning of the partial structure ([0062] & Fig. 5); and

a similar-pattern detecting unit that uses the similar structure generated by the similar-structure generating unit as an equivalent class of the partial structure on the generation source and detects the frequent pattern ([0060] & Fig. 3 frequency) by ignoring the difference between the attribute values in accordance with the second determination item of the similar-structure determination adjustment unit ([0062] & Fig. 5).

However, Kumai fails to teach a similar-structure generation adjustment unit that generates a first determination item for determining, from a user input,

a similar-structure determination adjustment unit that generates a second determination item for determining, from a user input,

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate a similar-structure generation adjustment unit that generates a first determination item for determining, from a user input and a similar-structure determination adjustment unit that generates a second determination item for determining, from a user input as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 9 and 20, Kumai teaches a text mining apparatus ([0008]) according to Claim 8, wherein the analyzing unit analyzes the document ([0062] & Fig. 5), and generates the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination,

an the attribute value includes the surface case and/or the information about the attached word, added to the sentence structure ([0062] & Fig. 5).

However, Kumai fails to teach generating the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text.

The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate generating the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination, as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 10 and 21, Kumai teaches a text mining apparatus according to Claim 8, wherein the similar-pattern detecting unit detects a frequent similar pattern ([0060] and [0062] & Fig. 3 and 5).

Re claims 11 and 22, Kumai teaches a text mining apparatus ([0008]) according to Claim 8, wherein the similar- structure generating unit comprises:

means for performing parallel modification of the sentence structure when the first determination item determines the parallel modification ([0062] & Fig. 5);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary when the first determination item includes replacement of the synonym ([0062] & Fig. 5); and the similar-structure generating unit generates a similar structure of the sentence structure and sets the similar structure as the equivalent class ([0062] & Fig. 5)

However, Kumai fails to teach means for generating the partial structure of the sentence structure; means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure when the first determination item determines the non-directional branching of the directional branch;

means for performing non-ordering of ordering trees of the sentence structure and/or partial structure when the first determination item determines the non-ordering of the ordering trees, and wherein;

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for generating the partial structure of the sentence structure; means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure when the first determination item determines the non-directional branching of the directional branch and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure when the first determination item determines the non-ordering of the ordering trees, and wherein as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. US 20030204496 A1, US 20040064447 A1, US 6272455 B1.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C. Colucci whose telephone number is (571)-270-1847. The examiner can normally be reached on 9:30 am - 6:00 pm, Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on (571)-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael C Colucci/
Examiner, Art Unit 2626
Patent Examiner
AU 2626
(571)-270-1847
Examiner FAX: (571)-270-2847
Michael.Colucci@uspto.gov

/Richemond Dorvil/
Supervisory Patent Examiner, Art Unit 2626